The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

An Optimal Footprint Method for Case-Base Maintenance

Ditty Mathew, Sutanu Chakraborti
Department of Computer Science and Engineering
Indian Institute of Technology, Madras
Chennai 600 036, India
{ditty, sutanuc}@cse.iitm.ac.in

Abstract

In Case-Based Reasoning (CBR), new problems are solved by retrieving similar previously solved cases and adapting their solutions. The new case is then stored appropriately in the case-base for future use. It is a fundamental problem to control the growth of case-base and the case-base maintenance step retains cases in the case-base based on an estimate of their usefulness in solving new problems. We propose an optimization formulation to identify an optimal set of representative cases called the optimal footprint of the case-base. The optimization formulation ensures that the optimal footprint set strikes a right trade-off between minimizing the number of cases and maximizing their ability to solve the remaining cases in the case-base. This trade-off is studied empirically in this paper. We also illustrate the trade-off between the size and performance of optimal footprint in the context of regression.

Introduction

Case-Based Reasoning (CBR) (Kolodner 1992) is an experience based learning methodology, which reuses past experiences to solve problems in future. It solves new problems by retrieving and adapting solutions of similar previously solved problems that have been stored in a repository called the case-base (De Mantaras et al. 2005). The case-base contains problem-solution pairs of problems that are solved in the past. For example, in regression data, each data instance corresponds to the problem and its target value corresponds to the solution. Each problem-solution pair is considered as a case in the case-base. The case-base size increases when more previously solved cases are added to the case-base. The size reduction of a case-base is ensured during the *Case-Base Maintenance* (Reinartz, Iglezakis, and Roth-Berghofer 2001, Smyth 1998) step, which retains cases in the case-base based on its quality to arrive at a solution for new problems. Competence of a case-base (Smyth and McKenna 1998) is the range of target problems that can be solved by the cases in that case-base. The footprint based approach (Smyth and McKenna 1999) is a competence guided case-base maintenance method to estimate a subset of cases in the case-base called the footprint set, which can solve the remaining cases in the case-base. More precisely, the footprint based approach is a data reduction approach in CBR which identifies a set of representative cases that can be retained in the case-base to find a solution for new problems.

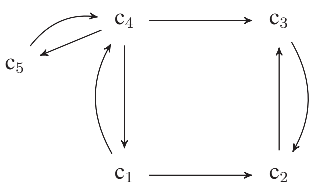

Figure 1: A sample case-base network

The footprint approach uses a greedy algorithm to estimate the footprint set that can solve the remaining cases in the case-base. Though this approach estimates a footprint set with size close to minimum, there is no guarantee for it. These footprint cases are estimated based on their ability to find a solution for a large range of cases in the case-base. However, the extent to which the footprint cases solve the remaining cases is not considered. Hence, the usage of footprint set as a surrogate of the original case-base can adversely affect CBR effectiveness. Mathew and Chakraborti (2017) propose a generalized case competence model that estimates the footprint set based on the extent to which other cases are being solved by the footprint set. Here the objective is to arrive at a footprint set that can solve all cases in the case-base more effectively. However, this approach also does not guarantee a minimal footprint set. In order to address the limitations of these existing approaches in literature, we attempt an investigation into an approach to arrive at an optimal footprint set while minimizing performance loss compared to the original case-base.

As in Mathew and Chakraborti (2017), we use the term *problem solving ability*, which is defined for a case c to solve another case t, to indicate the extent to which the case c is able to arrive at a solution for t. The loss in the performance of footprint set compared to the original case-base depends on the ability of cases in the footprint set to solve the rest of the cases in the case-base. We propose a convex optimization formulation to identify a footprint set with minimum size and maximum overall problem solving ability. We also study the trade-off between the footprint size and the performance loss using the proposed optimization method.

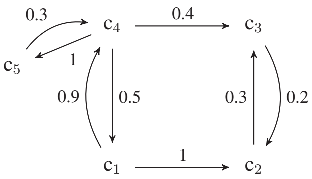

Figure 2: A sample case-base network with problem solving ability as weights

Cases	Relative Coverage	Weighted Retention Score
c_1	1	2
c_2	0.5	1.02
c_3	0.5	1
c_4	2.5	1.8
c_5	0.5	1.07

Table 1: Relative Coverage and Weighted Retention Score of Cases in Figure 1 and 2 respectively

Background

Footprint Based Approach

The footprint based approach identifies a set of representative cases that can be retained in the case-base, which can be used to arrive at a solution for the rest of the cases in the case-base. This method is a competence guided model where the competence of a CBR system is the range of target problems that the given system can solve. The footprint based approach follows a two step procedure to identify the footprint set or representative cases. Firstly, it orders cases in a decreasing order based on a measure called *relative coverage* (Smyth and McKenna 1999) which captures the individual contribution of each case in solving other cases in the case-base. Secondly, it identifies the footprint cases by following a greedy approach which processes cases in the relative coverage order, and each case is added to the footprint set only if the current footprint cannot solve it.

The idea of relative coverage is based on local competence properties such as *coverage* and *reachability* (Smyth and McKenna 1998). For the purpose of defining these properties, a relation *solves* (Smyth and McKenna 1998) is defined as

Definition 1. *solves(c,t): A case c solves a target problem t, if and only if c can be retrieved and adapted to obtain a solution for t*

Let $\mathbb{C}$ be the set of cases in the case-base. The coverage and reachability are defined as

Definition 2. *Coverage of a case c is a set of cases that are solved by c.*
$Coverage(c) = \{c_i \in \mathbb{C} | solves(c, c_i)\}$

Definition 3. *Reachability of a case c is a set of cases that can solve c*
$Reachability(c) = \{c_i \in \mathbb{C} | solves(c_i, c)\}$

For example, consider the case-base network given in Figure 1. The vertices denote cases and edges are drawn based on the definition of $solves$ in Definition 1, i.e., an edge (c_i, c_j) denotes that c_i can arrive at a solution for c_j. In Figure 1, the coverage of c_1 includes c_2 and c_4 and the reachability of c_2 includes c_1 and c_3.

The relative coverage measures the global competence of a case-base and it is defined as

$$RelativeCoverage(c) = \sum_{c_i \in Coverage(c)} \frac{1}{|Reachability(c_i)|} \qquad (1)$$

If a case c_i can be solved by n other cases in the case-base, then each of the n cases gets a contribution of $\frac{1}{n}$ from c_i to their relative coverage measures. It may be noted that the elegance of the footprint approach is that it encompasses all four knowledge containers in Case-Based Reasoning (Richter and Weber 2016) (the definition of solves above, for instance entails use of similarity, case-base, vocabulary and adaptation knowledge), and hence is ideally suited as our baseline.

The relative coverage of cases in Figure 1 are listed in Table 1. To identify the footprint set, the cases are processed in the descending order of their relative coverage. As the cases are processed based on the relative coverage order, highly competent cases get added to the footprint set before the less competent cases. In this example, c_4 has the highest relative coverage, which is added first to the footprint set. Since it solves c_1, c_3 and c_5, these cases are not required to be included in the footprint set. Finally, the footprint set evaluates to $\{c_4, c_2\}$. We note that the approach is greedy. Though cases covered by cases in the footprint set are excluded, there is no guarantee that the footprint set size is minimum.

Generalized Case Competence Model

The generalized case competence model (Mathew and Chakraborti 2017) handles CBR applications which involve single case or compositional adaptation (Wilke and Bergmann 1998). In the former applications, the solution of a target problem can be adapted from the solution of a single case, and in the latter one, the solution of the target problem can be adapted from solutions of multiple cases. However, the relative coverage based model (Smyth and McKenna 1999) handles only single case adaptation. This model uses a measure called *Weighted Retention Score* instead of *Relative Coverage* in Smyth and McKenna (1999) to estimate the footprint set. The weighted retention score estimates the retention quality of cases in the case-base and considers problem solving ability, i.e., the ability of cases in solving other cases while estimating the retention quality. This score is measured using a recursive formulation like PageRank (Page et al. 1999) and it is derived based on the intuition that the weighted retention score of a case c is high if

(a) c can find a solution for many cases with high problem solving ability

(b) the cases that can be solved by c have high weighted retention score

(c) c needs the support of less number of cases to find the solution for the cases that it solves. (a case may require the support of other cases to obtain the solution of a target case in compositional adaptation applications)

(d) the minimum of the weighted retention score of those cases that support c is high

A more detailed formulation of the weighted retention score can be found in Mathew and Chakraborti (2017).

Consider the case-base network given in Figure 2, this network is similar to Figure 1 except that it is weighted. The weight of an edge (c_i, c_j) denotes an estimate of the ability of c_i to arrive at a solution for c_j, i.e., the problem solving ability of c_i to solve c_j. The weights are normalized between 0 and 1 (inclusive). The weighted retention score values for all cases in the network are given in Table 1. To estimate the footprint set, like in Smyth and McKenna (1999) cases are sorted in the descending order of their weighted retention score. Then, while processing cases one by one in this order, a case is added to the footprint set if it cannot be solved by the cases in the footprint set. Thus, we obtain the footprint set as $\{c_1, c_5, c_3\}$. The problem solving ability of cases in the footprint set are high. The cases c_2 and c_4 are the cases that are not present in the footprint set; c_2 and c_4 can be solved by c_1 with problem solving ability 1 and 0.9 respectively. However, the footprint set size need not be minimal.

The work reported in this paper is driven by the intuition that *it may be interesting to explore a middle ground between minimizing the footprint size and maximizing the problem solving ability of the footprint cases*. The goal is to arrive at a general formulation of the case-base maintenance problem that can be tweaked to address application specific needs. It may also be noted that the choice of footprint strategy as the baseline ensures that our approach encompasses all the four knowledge containers of Case-Based Reasoning (Richter and Weber 2016) as noted earlier in this section.

Problem Statement

Let $\mathbb{C}$ be a set of cases in a case-base. We define a matrix P of dimension $|\mathbb{C}| \times |\mathbb{C}|$ to store values that correspond to the problem solving ability of all pairs of cases in $\mathbb{C}$. For each pair of cases $(c_i, c_j) \in \mathbb{C}$, we define the problem solving ability ($P(c_i, c_j)$) as the extent to which the case c_i is able to arrive at a solution of c_j and the matrix P is characterized as,

1. $P(c_i, c_i) = 1$
2. $0 \leq P(c_i, c_j) \leq 1$
3. $P(c_i, c_j) = 0$ if c_i does not solve c_j

The goal is to estimate an optimal footprint set $FP_{opt} \subseteq \mathbb{C}$ as representatives such that

1. FP_{opt} can solve all cases in $\mathbb{C}$ with high solution quality
2. FP_{opt} size is minimal

Problem Formulation

We formulate a convex minimization problem to estimate the optimal footprint set FP_{opt} from a case-base $\mathbb{C}$.

Consider the binary vector $x =< x_{c_i} >_{c_i \in \mathbb{C}}$, where each element x_{c_i} corresponds to a case $c_i \in \mathbb{C}$ and it indicates whether the case c_i is present in the optimal footprint set or not, i.e.,

$$x_{c_i} = \begin{cases} 1 & \text{if } c_i \in FP_{opt}, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Let $loss =< loss_{c_i} >_{c_i \in \mathbb{C}}$ be a loss vector of a case-base with respect to the footprint set (FP_{opt}). Let c_j be the case in FP_{opt} that solves a case $c_i \in \mathbb{C}$ with a maximum problem solving ability $P(c_j, c_i)$. Then the $loss$ of a case c_i with respect to FP_{opt} is defined as $1 - P(c_j, c_i)$. If a case c_i can be solved by more than one case in FP_{opt}, then the case with the maximum problem solving ability is used for loss computation. $loss_{c_i}$ is zero if $c_i \in FP_{opt}$. The loss function is formally defined as,

$$loss_{c_i} = 1 - \max_{c_j \in \mathbb{C}} P(c_j, c_i) * x_{c_j} \tag{3}$$

Let $Cov =< Cov_{c_i} >_{c_i \in \mathbb{C}}$ be a vector, which indicates whether a case c_i can be solved by the footprint FP_{opt} or not. Mathematically, it is defined as,

$$Cov_{c_i} = \begin{cases} 1 & \text{if } c_i \text{ can be solved by } FP_{opt} \\ 0 & otherwise \end{cases} \tag{4}$$

We formulate an optimization problem as a Mixed Integer Program (Wolsey 2008) where the objective function is to minimize the footprint size and the overall loss of all cases in the case-base, and the footprint set is constrained to solve all cases in the case-base. The objective function and constraints are given in Equations 5a, 5b and 5c respectively.

$$\begin{aligned} \text{min} \quad & \sum_{c_i \in \mathbb{C}} (loss_{c_i} + x_{c_i}) & \text{(5a)} \\ \text{subject to} \quad & \sum_{c_i \in \mathbb{C}} Cov_{c_i} = |\mathbb{C}|, & \text{(5b)} \\ & x_{c_i} \in \{0, 1\} \qquad \forall 1 \leq i \leq n. & \text{(5c)} \end{aligned}$$

All constraints are linear, hence they are convex. However, the loss function is concave due to the $\max$ term in it. This makes the objective function concave and the optimization problem concave minimization problem. Hence this problem cannot be solved as a convex optimization problem. The equivalent linear function of $\max$ can be obtained by using a binary variable (Fico 2009). For example, suppose we want to find the $\max(v_1, v_2, \ldots, v_n)$ where $0 \leq v_i \leq 1$ for $1 \leq i \leq n$. The linear function of $\max$ introduces a new variable y and a binary variable d of dimension n. Let $y = \max(v_1, v_2, \ldots, v_n)$, then $y \geq v_i \;\; \forall 1 \leq i \leq n$. These constraints find a y such that,

$$y \geq \max(v_1, \ldots, v_n) \tag{6a}$$

The binary variable d is used for finding an upper bound for y and these constraints are as follows.

$$\begin{aligned} & y \leq v_i + 1 - d_i \qquad \forall 1 \leq i \leq n, & \text{(7a)} \\ & \sum_{i=1}^{n} d_i = 1, & \text{(7b)} \\ & d_i \in \{0, 1\} \qquad \forall 1 \leq i \leq n, & \text{(7c)} \end{aligned}$$

The constraints 7b and 7c ensure that only one value of d is 1 and remaining are 0. The v_i value corresponds to $d_i = 1$ acts as the upper bound for y variable as per constraint 7a. These constraints and the constraint 6a find a feasible solution only when $d_i = 1$ for the v_i with maximum value. This results in the bounded constraints $y \geq \max(v_1, \ldots, v_n)$ and $y \leq \max(v_1, \ldots, v_n)$. Using this idea, the equivalent linear function corresponds to the one defined in Equation 3 is,

$$loss_{c_i} = 1 - y_i \tag{8}$$

Method	FP	Total loss	Objective value
FP_{opt}	c_1, c_4	0.6	2.6
FP_{rc}	c_4, c_2	1.1	3.1
FP_{wrs}	c_1, c_5, c_3	0.1	3.1

Table 2: Comparison of footprint size and total loss for the example in Figure 2

where y_i is constrained as in the revised optimization formulation given below.

$$\min \quad \sum_{c_i \in \mathbb{C}} (1 - y_i + x_{c_i}) \tag{9a}$$

$$\text{subject to} \quad \sum_{c_i \in \mathbb{C}} Cov_{c_i} = |\mathbb{C}|, \tag{9b}$$

$$x_{c_i} \in \{0, 1\} \quad \forall 1 \leq i \leq n, \tag{9c}$$

$$y_i \geq P(c_j, c_i) * x_{c_i} \quad \forall 1 \leq i, j \leq n, \tag{9d}$$

$$y_i \leq P(c_j, c_i) * x_{c_i} + 1 - d_j \quad \forall 1 \leq i, j \leq n, \tag{9e}$$

$$\sum_{i=1}^{n} d_i = 1, \tag{9f}$$

$$d_i \in \{0, 1\} \quad \forall 1 \leq i, \leq n. \tag{9g}$$

The constraints 9d, 9e, 9f, and 9g ensure that y_i takes the maximum value of $P(c_j, c_i) * x_{c_i}$ for all $1 \leq i \leq n$.

For example, consider a sample case-base network given in Figure 2, which contains cases $\mathbb{C} = \{c_1, c_2, c_3, c_4, c_5\}$. Using weights of edges in the network, its problem solving ability matrix P is obtained as,

$$P = \begin{bmatrix} 1 & 1 & 0 & 0.9 & 0 \\ 0 & 1 & 0.3 & 0 & 0 \\ 0 & 0.2 & 1 & 0 & 0 \\ 0.5 & 0 & 0.4 & 1 & 1 \\ 0 & 0 & 0 & 0.3 & 1 \end{bmatrix}$$

FP_{opt} is estimated based on the proposed optimization formulation. The footprint size and the summation of the loss of all cases in the case-base (total loss) are compared with those from the original footprint approach (Smyth and McKenna 1999) which uses relative coverage (FP_{rc}) and also with the footprint set estimated based on weighted retention score (FP_{wrs}) (Mathew and Chakraborti 2017). The comparison is given in Table 2.

Though the size of FP_{opt} and FP_{rc} are same, both footprint sets are different and the total loss of FP_{opt} is much less than FP_{rc}. The total loss of FP_{wrs} is much less compared to the other two approaches, whereas the size is not minimum.

In constraint 9b, the Cov_{c_i} is estimated based on the involvement of any case in FP_{opt} in solving the case c_i. However, the extent to which the case c_i is being solved by FP_{opt} is not considered. The soft definition of Cov_{c_i} as in Equation 4 can be redefined as

$$Cov_{c_i} = \begin{cases} 1 & \text{if } c_i \text{ can be solved by } c_j \in FP_{opt} \text{ with } P(c_j, c_i) \geq \beta \\ 0 & otherwise \end{cases} \tag{10}$$

The parameter β decides the threshold that dictates whether the case c_i can be considered to be solved by the case $c_j \in FP_{opt}$. This redefinition of Cov_{c_i} will reduce the loss whereas it will increase the size of footprint set. For example, let $\beta = 0.5$; consider $FP_{opt} = \{c_1, c_4\}$ that is obtained for the example in Figure 2. Then, the Cov vector with respect to $\beta = 0.5$ as per Equation 10 is {1, 1, 0, 1, 1}. As $Cov_{c_3} = 0$, the constraint 9b is not valid. Hence, the optimal footprint set based on the revised definition is $\{c_1, c_3, c_4\}$. The loss corresponding to this footprint set is zero.

Dataset	# instances	# features
housing	506	13
auto MPG	392	7
hardware	209	7
automobile	194	12

Table 3: Dataset characteristics

Dataset	FP_{opt}		FP_{rc}		FP_{wrs}	
	size	loss	size	loss	size	loss
housing	396	23.09	397	24.78	464	8.56
auto MPG	302	19.36	303	22.04	345	8.4
hardware	157	15.24	156	16.24	165	11.65
automobile	114	4.5	115	5.3	121	4.5

Table 4: Comparison of footprint size and loss over different datasets

Experiments

We evaluate the proposed method on four datasets: housing, auto MPG, computer hardware, and automobile. These datasets are available in UCI Repository (Bache and Lichman 2013). The goal of these datasets is to predict the housing price, fuel consumption, estimated relative performance, and automobile price respectively. The data instances with unknown values are removed from all datasets, non-numeric features are omitted, and feature values are normalized between 0 and 1. The characteristics of all datasets are summarized in Table 3.

To illustrate the proposed optimal data reduction technique, we consider each dataset as a case-base where the data instances are cases. We use the nearest neighbor algorithm (Cover and Hart 1967) to identify the nearest case that can predict the target value of each case while keeping the acceptable prediction error fixed at 5%. A case-base network is constructed with cases as vertices and each directed edge

Dataset	FP_{rc}	FP_{wrs}	FP_{opt}		
			$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
housing	27.46	27.47	27.61	**27.35**	25.63
auto MPG	12.02	**11.5**	12.18	11.54	10.93
hardware	45.55	45.55	**45.5**	**45.5**	45.45
automobile	6.49	6.52	6.38	**6.23**	5.92

Table 5: Mean Square Error of test data when trained with footprint sets FP_{rc}, FP_{wrs} and FP_{opt} with $\alpha = 0, 0.5, 1$

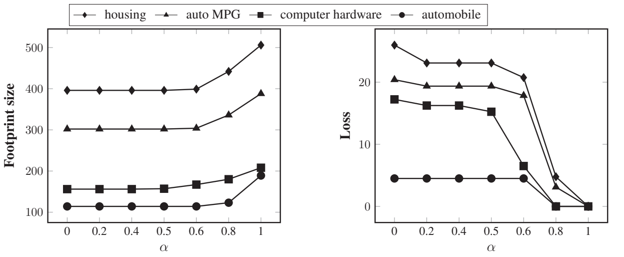

Figure 3: Trade-off between footprint size and loss for all datasets

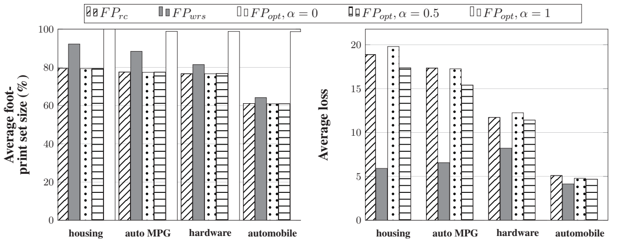

Figure 4: Average size and loss of footprint sets (FP_{rc}, FP_{wrs}, FP_{opt} with $\alpha = 0, 0.5, 1$) obtained from 5-fold training data

(u, v) in the network denotes the case u can predict the target value of the case v with an error percentage less than or equal to 5%. Each edge (u, v) is associated with a weight that corresponds to the problem solving ability of u to solve v. The problem solving ability of a case c to solve the problem t ($P(c, t)$) is measured as,

$$P(c, t) = \frac{1}{1 + (y_{actual} - y_{predict})^2} \tag{11}$$

where y_{actual} is the actual solution of t and $y_{predict}$ is the estimated solution predicted by c for t. We estimate the footprint set FP_{opt} from the case-base network using the proposed optimization formulation. FP_{opt} is compared with the footprint sets obtained using greedy algorithms such as relative coverage based footprint (FP_{rc}) and weighted retention score based footprint (FP_{wrs}). In Table 4, the footprint size and overall loss of FP_{opt}, FP_{rc}, and FP_{wrs} are given for all four datasets.

We can observe that the footprint size of FP_{opt} and FP_{rc} are almost same. However, the loss of FP_{opt} is less compared to FP_{rc} for all datasets. Although the loss of FP_{wrs} is much less compared to other two footprint sets, the footprint size is much more than others. This is because FP_{wrs} finds a footprint set with high problem solving ability by compromising on footprint size. FP_{rc} almost reaches the optimal footprint size, but the loss is not minimized. FP_{opt} identifies a footprint set which balances both footprint size and loss.

Dataset	$\beta = 0.2$	$\beta = 0.4$	$\beta = 0.6$	$\beta = 0.8$	$\beta = 1$
housing	27.35	27.35	25.82	25.82	25.63
auto MPG	11.54	11.50	11.50	11.31	11.10
hardware	45.49	45.47	45.45	45.45	45.45
automobile	6.23	6.23	6.0	6.0	6.0

Table 6: Mean Square Error of test data when trained with FP_{opt} with $\alpha = 0.5$ and $\beta = 0.2, 0.4, 0.6, 0.8, 1$

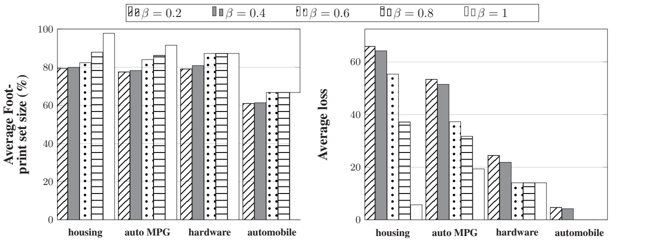

Figure 5: Average size and loss of FP_{opt} sets with $\alpha = 0.5$ and at $\beta = 0.2, 0.4, 0.6, 0.8, 1$ estimated from 5-fold training data

Trade-off between footprint set size and loss

We analyze the trade-off between the footprint set size and the loss by changing the objective function into a weighted sum of both size and loss. The modified objective function is given as,

$$\min \qquad \alpha * \sum_{c_i \in \mathbb{C}} loss_{c_i} + (1 - \alpha) * \sum_{c_i \in \mathbb{C}} x_{c_i} \tag{12}$$

where $0 \leq \alpha \leq 1$. When $\alpha = 0$, the objective function minimizes only based on size and when $\alpha = 1$, the objective function relies only on loss. When $\alpha = 0.5$, both footprint size and loss are given same importance. The trade-off is illustrated in Figure 3.

For all datasets, we can observe that the footprint size remains constant from $\alpha = 0$ till $\alpha = 0.5$ and it slightly increases at $\alpha = 0.6$. After that when $\alpha > 0.6$, the footprint size grows relatively rapidly and the size reaches close to the original case-base size at $\alpha = 1$. Despite the footprint size remaining constant from $\alpha = 0$ till $\alpha = 0.5$, the loss decreases very slowly for these α values. The loss decreases fast from $\alpha = 0.6$ with increase in the footprint size and loss is 0 at $\alpha = 1$. For the computer hardware dataset, there is a sudden decrease in loss after $\alpha = 0.5$, and for housing and auto MPG dataset this effect is seen after $\alpha = 0.6$. This is because the footprint size increased rapidly after the corresponding α values. Although the footprint sizes of small datasets such as the computer hardware and automobile increase from $\alpha = 0.8$ to $\alpha = 1$, the losses reach zero at $\alpha = 0.8$. This indicates that the compressed footprint set at $\alpha = 0.8$ performs equivalent to the one at $\alpha = 1$.

Footprint set performance analysis

The performance of the optimal footprint set is analyzed by using the footprint set as training data in regression applications. For each dataset, we take 80% of data instances as training data and 20% as test data. The footprint set is estimated from the training data and is used as training data to predict the target value of the test data. We perform 5-fold cross validation and the average of the mean square error (MSE) of the predicted values are used for analyzing the performance of the footprint set. We experiment with footprint set based on relative coverage (FP_{rc}), footprint set based on weighted retention score (FP_{wrs}), and optimal footprint set (FP_{opt}) based on Equation 12 with $\alpha = 0$, $\alpha = 0.5$, and $\alpha = 1$. The size of footprint sets obtained based on each method from 5-fold training data are averaged and the corresponding average loss is compared in all datasets and reported in Figure 4. We can observe that the footprint set size of FP_{rc} and FP_{opt} with $\alpha = 0$ and $\alpha = 0.5$ are almost the same. However, the average loss of FP_{opt} with $\alpha = 0.5$ is less compared to FP_{rc} and FP_{opt} with $\alpha = 0$. FP_{opt} with $\alpha = 1$ adds almost all cases to the footprint set due to which its size is close to 100% and its loss is close to zero. The size and loss of FP_{wrs} lie in between the size and loss of FP_{opt} with $\alpha = 0.5$ and $\alpha = 1$.

The performance of each footprint set is evaluated based on the mean square error (MSE) of test data while the corresponding footprint set acts as the training data. The mean square error obtained for all footprint sets when experimented in all four datasets are given in Table 5. FP_{opt} with $\alpha = 1$ is almost same as the original training data, hence its MSE is the lowest compared to others in all datasets. FP_{opt} with $\alpha = 0.5$, scores the next lowest MSE in all datasets except auto MPG for which FP_{wrs} scores second lowest. FP_{opt} with $\alpha = 0.5$ compresses more than FP_{wrs} and performs close to FP_{opt} with $\alpha = 1$ (i.e., original training data) compared to other footprint sets. Among the footprint sets - FP_{rc}, and FP_{opt} with $\alpha = 0$ and $\alpha = 0.5$ that have high compression rate, FP_{opt} with $\alpha = 0.5$ scores the lowest MSE in all datasets.

We also analyze the FP_{opt} sets with different β values as per Equation 10. We fixed $\alpha = 0.5$ and obtained FP_{opt} with $\beta = 0.2, 0.4., 0.6, 0.8, 1$. These footprint sets are used as training data and the mean square error obtained over test data are analyzed and the error values are given in Table 6. The corresponding footprint set size and its loss are given in Figure 5. We can observe that the MSE of footprint sets decreases with increase in the value of β, the average size of footprint set increases as β increases, and this results in decrease in average loss as β increases. From $\beta = 0.6$ till $\beta = 1$, the MSE of FP_{opt} sets are close to the MSE of FP_{opt} with $\alpha = 1$ as in Table 5. The parameters α and β can be chosen according to the extent to which the footprint set needs to be compressed and the loss needs to be reduced in the domain under consideration.

Conclusion and Future Work

We propose an optimization formulation to estimate a set of representative cases called footprint set from a case-base, which optimizes based on the size and ability of footprint cases to solve the remaining cases in the case-base. We perform a trade-off analysis between the footprint set size and the problem solving ability in four datasets. The trade-off between performance of footprint set and its size is also studied.

The current optimization formulation assumes single case adaptation. We would like to extend the formulation to apply this approach in compositional adaptation applications (Mathew and Chakraborti 2016).

Acknowledgments

We thank Devi Ganesan for helpful discussions and feedback, and the members of AIDB lab for their insightful comments. We also gratefully acknowledge the support from Tata Consultancy Services (TCS) through TCS Research Scholar Program.

References

Bache, K., and Lichman, M. 2013. UCI Machine Learning Repository.

Cover, T. M., and Hart, P. E. 1967. Nearest Neighbor Pattern Classification. *IEEE Transactions of Information Theory* 13(1):21–27.

De Mantaras, R. L.; McSherry, D.; Bridge, D.; Leake, D.; Smyth, B.; Craw, S.; Faltings, B.; Maher, M. L.; T COX, M.; Forbus, K.; et al. 2005. Retrieval, Reuse, Revision and Retention in Case-Based Reasoning. *The Knowledge Engineering Review* 20(3):215–240.

Fico. 2009. Mip Formulations and Linearizations. Technical report, Fair Isaac Corporation.

Kolodner, J. L. 1992. An Introduction to Case-Based Reasoning. *Artif. Intell. Rev.* 6(1):3–34.

Mathew, D., and Chakraborti, S. 2016. Competence Guided Casebase Maintenance for Compositional Adaptation Applications. In *Proceedings of International Conference on Case-Based Reasoning*, 265–280.

Mathew, D., and Chakraborti, S. 2017. A Generalized Case Competence Model for Casebase Maintenance. *AI Communications* 30(3-4):295–309.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Reinartz, T.; Iglezakis, I.; and Roth-Berghofer, T. 2001. Review and Restore for Case-Base Maintenance. *Computational Intelligence* 17(2):214–234.

Richter, M. M., and Weber, R. O. 2016. *Case-Based Reasoning*. Springer.

Smyth, B., and McKenna, E. 1998. Modelling the Competence of Case-bases. In *Proceedings of European Workshop on Advances in Case-Based Reasoning*, 208–220.

Smyth, B., and McKenna, E. 1999. Footprint-Based Retrieval. In *Proceedings of International Conference on Case-Based Reasoning*, 343–357.

Smyth, B. 1998. Case-Base Maintenance. In *Proceedings of International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, 507–516.

Wilke, W., and Bergmann, R. 1998. Techniques and Knowledge used for Adaptation during Case-Based Problem Solving. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 497–506.

Wolsey, L. A. 2008. Mixed Iinteger Programming. *Wiley Encyclopedia of Computer Science and Engineering*.